

# Cognitive Coordination of Global Service Delivery

Lav R. Varshney, Shivali Agarwal, Yi-Min Chee, Renuka R. Sindhgatta,  
Daniel V. Oppenheim, Juhnyoung Lee, and Krishna Ratakonda



**Abstract**—Formal coordination mechanisms are of growing importance as human-based service delivery becomes more globalized and informal mechanisms are no longer effective. Further it is becoming apparent that business environments, communication among distributed teams, and work performance are all subject to endogenous and exogenous uncertainty.

This paper describes a stochastic model of service requests in global service delivery and then puts forth a cognitive approach for coordination in the face of uncertainty, based on a perception-action loop and receding horizon control. Optimization algorithms used are a mix of myopic dynamic programming and constraint-based programming. The coordination approach described has been deployed by a globally integrated enterprise in a very large-scale global delivery system and has been demonstrated to improve work efficiency by 10 – 15% as compared to manual planning.

## 1 INTRODUCTION

With the emergence of systems that bring together ubiquitous information technologies with the people and organizations they are transforming, it is important to understand how to direct and coordinate so as to achieve optimal efficiency. Firms are perhaps the most sophisticated of such sociotechnical systems, where people come together to develop innovative products and services. The *global service delivery* approach to doing knowledge work requires coordinating tens of thousands of specialized workers distributed around the world and has become prominent in many enterprises. Handling such large-scale agglomerations of people and machines, however, requires developing new abstractions, approaches, and algorithms. This paper explicates the practical design of one such sociotechnical system for global service delivery and cognitive methods of coordination within it. A key aspect of system design is to understand and model the diversity of humans, and their preferences.

*This work is based in part on a paper presented at the 12th International Research Symposium on Service Excellence in Management (QUIS12) [1]. L. R. Varshney was with the IBM Thomas J. Watson Research Center. He is now with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL (e-mail: varshney@illinois.edu).*

*S. Agarwal and R. Sindhgatta are with the IBM India Research Laboratory, Bangalore, India (e-mail: {shivaaga, renuka.sr}@in.ibm.com).*

*Y.-M. Chee, D. V. Oppenheim, J. Lee, and K. Ratakonda are with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY (e-mail: {ymchee, music, jyl, ratakond}@us.ibm.com).*

Whether engaged in designing a physical system like an airplane or building an information system like customer relationship management software, organizations providing informational services are becoming more and more globalized with an increasing degree of workforce specialization [2]–[4]: specialized teams that concentrate on a narrow set of tasks can often be more productive than teams that are jacks-of-all-trades. Indeed, the tradeoff between productivity benefits provided by specialization and coordination costs incurred with a distributed workforce are well-known in economic theory [5], [6], but globalization makes the value of specialization through division of labor more important now than ever before [7].

Unfortunately project failures, excessive delays, and significant financial losses have been observed in many global service delivery projects. Traditional project management techniques for co-located teams such as mutual adjustment through informal communication [8] do not scale well to a global workforce [9]. Four main problems in global software development include [10]:

- Conflicts of interest arising due to distributed work teams with local incentives,
- Interdependencies arising from distributed work processes,
- Technology representation problems arising from distributed technologies with local standards, and
- Uncertainties and equivocalities arising due to geographically and organizationally distributed information.

Although resolving conflicts of interest is certainly important, the cooperative elements of global collaboration are distinct from the coordinative ones [11]. As part of designing our cognitive coordination for global service delivery, we aim to minimize occurrence of the last three problems.

The framework, approach, and algorithms detailed in the sequel arose from designing and implementing a new information technology framework for global service delivery: IBM's Application Assembly Automation (AAO), which has become a key component of IBM's Globally Integrated Capabilities [12]. Large software development projects that were once carried out



Fig. 1. Delivery centers (dots) and coordination hubs (stars) distributed around the world.

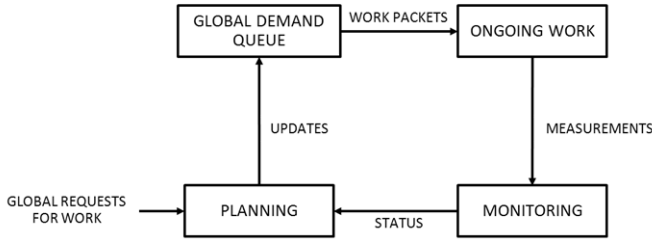


Fig. 2. Perception-action cycle for global service delivery.

by large colocated teams are now broken into pieces and executed in isolation by an interchangeable delivery center. Different delivery centers specialize in different aspects of software development, such as design, service-oriented architecture development, or testing, and are strategically located globally, as depicted in Figure 1. Each piece of work is routed to a delivery center through a construct called a *work packet*, and the overall deliverables are then integrated in coordination hubs. AAO has some similarities to other global delivery systems [13].

The basic cognitive coordination approach we develop herein employs a *perception-action loop* as a central construct, see Figure 2 for a block diagram representation. By perception-action loop, we mean the ability of a system to continuously monitor its own behavior and the environment and to react accordingly to achieve a goal. Such loops are not only useful for describing human cognition [14], but also for building cognitive dynamic systems [15]–[17]. Although cognitive coordination can certainly support human decision making [18], automated assignment of work to workers via task lists is considered here.

The coordination algorithms follow the principle of receding horizon control (RHC). In RHC, also known as model predictive control, an optimization problem is solved at each time step to determine a plan of action over a fixed time horizon. The first control action from this plan is applied to the system. Then, at the next time step the planning process is repeated with a new optimization problem created with the time horizon shifted one time step forward. Optimization takes uncertainties and estimates of future quantities based on available information into account at each time step [19]. The specific algorithm developed for computational

efficiency in optimizing the large-scale system is based on a Markov decision process (MDP) formulation, which yields a mix of myopic dynamic programming (which uses limited information) and of constraint-based programming (which uses heuristic stopping rules).

Cognitive coordination for global service delivery enables scaling to larger and larger numbers of workers carrying out more and more work, faster response to business needs, and greater visibility. It has also led to 10–15% improved quality and productivity based on initial findings.

## 2 GLOBAL SERVICE DELIVERY BASICS AND TECHNOLOGIES

The basic idea of global service delivery is to undertake several large service engagements with a globally distributed workforce.

Traditional approaches to distributed service delivery have used deterministic models of the business environment, of communication among people, and of the work itself to address interdependencies. This has led to standardized communication protocols and encapsulations of service work, as well as coordination mechanisms that deal with interdependencies using business process and business entity lifecycles [20]–[23]. Rather than strict business process management approaches, we use an instantiation of the *work-as-a-service* protocol and algebra [24], [25] to encapsulate work into pieces and define operations for its management; this flexible protocol is amenable to handling uncertainties that are inherent in human-intensive work.

Work-as-a-service formalizes various operations such as *merge*, *tear*, *pause*, and *resume* which allow the control actions we will need to develop the RHC algorithms [24]–[26]. We do not go into details of the work-as-a-service algebra in this paper, but focus on the larger optimized coordination enabled by it. Large service engagements can be decomposed into several smaller work packets, and conversely several work packets can be combined into larger work packets. These work packets can also be delegated and reassigned to other service providers without any global impact since they are self-contained and explicitly include dependency relationships. Since work packets of any level of specificity can be decomposed, delegated, and reassigned, any service engagement can be thought of as comprising several atomic service requests. A complete service engagement would have a (perhaps hierarchical) network of atomic service requests, each dealt with in the same manner (due to uniformity of work packets).

As part of developing a perception-action loop for global service delivery, it is important to understand timescales over which management and planning actions can be taken. In our view there are four such basic timescales with associated actions:

- 1) *scale of years*: High-level strategy such as which country to locate a work center in response to labor

markets, costs, etc., as well as high-level strategy on kinds of service work to pursue.

- 2) *scale of months*: Hiring new people and dropping current people in response to gaps/gluts, as well as decisions to pursue specific service engagements.
- 3) *scale of days/hours*: Assignment of work tasks to workers in response to needs, skills, synergies, and interdependencies.
- 4) *scale of minutes*: Ad hoc rejigging of work assignments in response to perturbations that cannot be dealt with through re-planning

Our main focus in this paper is on the day/hour scale and thus on assignment of work tasks to workers.

Besides the ability to act, it is important to measure what is going on within the system and in the external environment [27]. Moving from measurement definition to measurement collection in a global service delivery environment has oft been complicated. Moreover, there have been inconsistencies from project to project of what gets collected, how it gets collected, and when. Inconsistencies even occur in a given metric when measured across two executions of the same process because of individual human variation.

To address these measurement challenges, we introduced a metrics framework (detailed elsewhere [28]). The framework provides consistency and commonality across comparable measurements; the ability to define arbitrary levels of granularity of what is being measured; flexibility of changing metrics in the face of contextual changes; deep visibility at all levels; and automation.

Besides its central role in continuous monitoring of system state, the metrics framework has also been used to characterize the work itself (encapsulated in terms of the work packet algebra), as well as to characterize the people that carry it out. In particular, when developing measurements for global service delivery it is important to understand a skills assessment of people [29], an assessment of the work itself [30], as well as the social history of people [31]. Figure 3 illustrates how we use various measured data sources to inform optimization algorithms by understanding interdependencies in work, how well certain people are matched to certain tasks, and how well people work together.

In addition to knowing what can be perceived and what can be acted upon, we also need to define a mechanism for action. Following fruitful precedent in controlling large-scale systems [32], we employ middleware technology to act as a coordination hub between service requesters and service providers, as depicted in Figure 4. The goal of the middleware is to allow organizational modularity without incurring transaction costs [33]–[35]. As can be noted, requesters of work need not interact with the providers of work; all communication is routed through the coordination hub through formal mechanisms. The various information flows follow the work-as-a-service protocol.

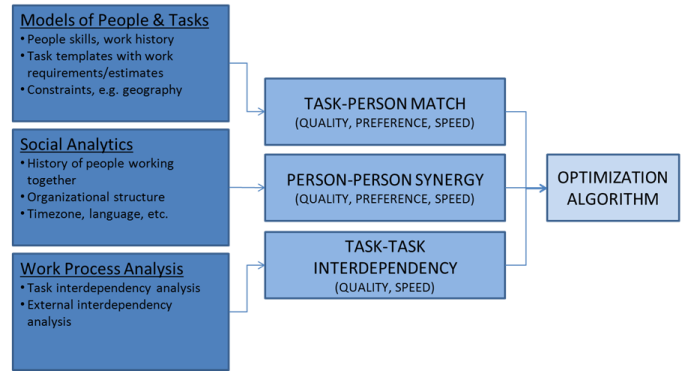


Fig. 3. Planning combines data from multiple sources to generate constraints/objectives for optimal assignments.

### 3 PERCEPTION-ACTION LOOP

Our approach to coordinating sociotechnical systems is cognitive and uses the principle of receding horizon control, yet is similar to other coordination mechanisms [36]. A schematic diagram of the perception-action loop for global service delivery is depicted in Figure 2. The main exogenous perturbation is the introduction of new global requests for work. These are given to the planning subsystem, which also has access to a real-time status signal from the monitoring subsystem. Work is buffered in the global demand queue and is dispatched as work packets to workers.

#### 3.1 Receding Horizon Control

RHC is a feedback control technique that became popular in the 1980s for physical systems [19], but does not seem to have previously been used for coordinating large-scale sociotechnical systems. Thinking of time proceeding in hour-long steps, with RHC, an optimization problem is solved at each time step to determine a plan of work assignment over a fixed time horizon thereafter. Optimization takes into account uncertainty and estimates of the future using available information at each time step.

Since global knowledge work is characterized by a high degree of unpredictability from human factors, complexity, size, changing requirements, and the environment itself, a stochastic model is necessary. The nonlinear coordination policy uses feedback from real-time measurements and handles input constraints, output constraints, and various control objectives.

Consider a work unit to be assigned as represented by a work breakdown structure such that each node in that structure graph represents a task to be assigned to one worker, and the edges represent interdependency constraints. There are many such units to be planned at a time. Work assignment is a matching problem where a work unit is assigned to an appropriately skilled resource so service delivery objectives are met. Existing work may get modified as time progresses and new

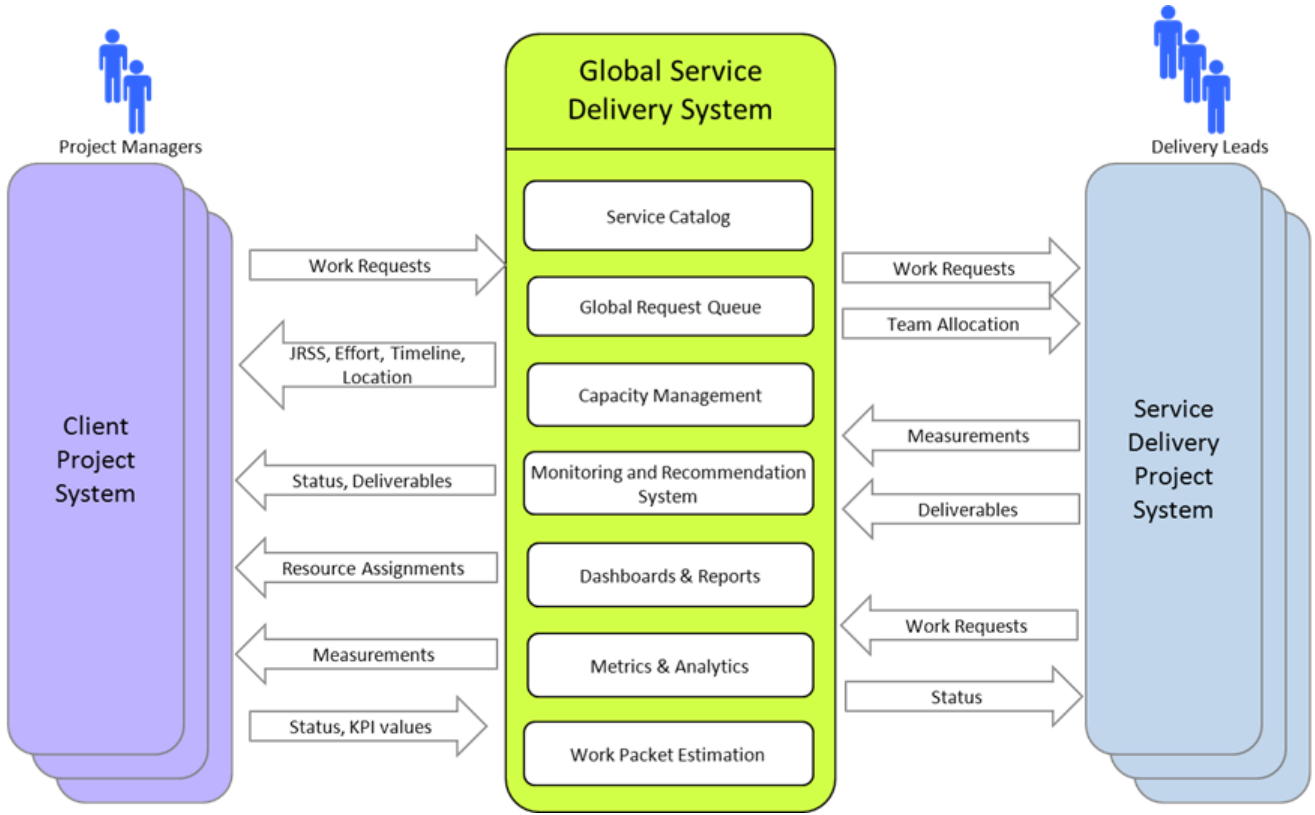


Fig. 4. Global service delivery system design, with a coordination hub acting between requests and delivery centers.

work keeps arriving. The work units may undergo modifications of various types like altering the work structure or effort estimate or desired start/completion times or the preferred resources/geography.

In addition to work dynamism, worker resources are also dynamic in terms of availability, skills, etc. A worker may go on leave and so work assigned to him may have to be reassigned. The worker may want to enhance his skills to eventually undertake different tasks than before. The worker's role may change, e.g. from developer to subject matter expert, which will necessitate reassignment of tasks planned assuming his previous role.

With this inherent dynamism at various timescales of control, RHC is useful for work assignment in global delivery systems. The need becomes even more pronounced due to the volume of work and the scale of the teams. The output of planning is a *work system plan* that is continuously updated in a palimpsestic manner as output of the planning subsystem in Figure 2. The work plan decision variables are actions at each timescale of action:

- *L1 decision*, e.g. start work center in Wisconsin next year;
- *L2 decision*, e.g. hire 13 Java programmers next month;
- *L3 decision*, e.g. assign task 347 to worker 872 to start tomorrow; and
- *Brownout decision*, e.g. reassign task 85 from worker 872 to worker 873 since worker 872 just got ill.

For psychological benefits, these decisions are translated into task lists before presentation to workers [37].

The remainder of this section discusses algorithmic approaches for dynamic optimization, focusing on L3 decisions: assigning work to workers.

### 3.2 Markov Decision Process Formulation

When focusing on L3 decisions (the assignment of work to workers) and certain aspects of L2 decisions on hiring/releasing workers, a Markov decision process (MDP) formulation is natural, since time proceeds in stages and there is a notion of state that captures all dependencies between past and future [38]. As depicted in Figure 5, at each stage there is an assignment to be made, along with a longer-term decision on whether to invoke “dummy resources” that correspond to resources that have not yet been hired. It is assumed that preemption is not allowed due to the loss of robustness and the inefficiency [25] it causes. Hence once a task is assigned to a worker, the state variable of that worker is set to the time remaining for the task.

The goal of a coordination algorithm is to optimize assignments within the MDP problem, however without preemption this is a computationally complicated integer program. A natural approach for finding the globally optimal solution would be via dynamic programming, but the state space is incredibly large. In our global delivery setting, we foresaw requirements of: (a) small-scale optimization over short timescales for every invo-



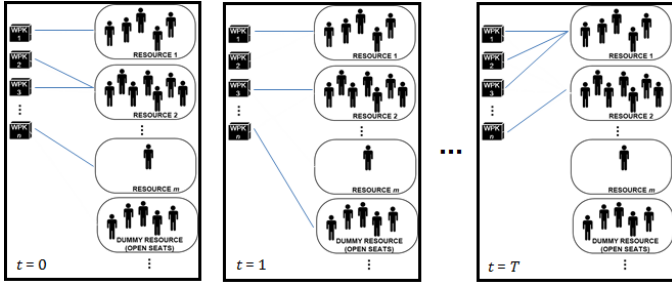


Fig. 5. Depicting the Markov decision process view of work assignment for global service delivery over time  $t$ . Note that the possible dummy resources can be added through L2 actions, whereas assignment is through L3 actions (blue lines).

cation almost at the time granularity of every deliverable arrival, and (b) approximate global optimization periodically so as to keep things on track. An RHC approach may therefore be effective.

Algorithmic development is built on the hypothesis that a myopic version of dynamic programming with a limited state space will be effective in cases of small and frequent perturbations whereas a constraint-based programming approach that uses global information heuristically<sup>1</sup> will be effective to find assignments given a fresh input set of large size.

We develop coordination that combines these two approaches: constraint-based programming uses global information but a heuristic optimization principle, whereas limited-horizon dynamic programming uses limited information but performs full optimization. Thus we end up with RHC that has stage-by-stage bipartite matching within the myopic dynamic programming and periodic globally optimal *scheduling checkpoints* via constraints. Though we omit formal statement and proof of this result, one can argue that it is nearly optimal to use the (optimal) Hungarian method for bipartite matching and constrained programming for (nearly) globally optimal checkpoints, by closely examining Bellman's equation. We present experiments to adjudicate performance in Section 4.

### 3.3 Constraint-Based Programming

To get near-optimal scheduling checkpoints that use information far into the future, we use constraint-based programming [39], [40]. Inputs are a set of work units and the pool of resources, recast as a set of constraints and objectives. The output of optimization is a complete schedule of the work as assigned on worker calendars. The main constraints from Figure 3 are as follows.

**Skill match:** To do work, some skills and attributes are mandatory whereas others are optional. For a government services programming task, Java skill and Amer-

ican geographical location may be mandatory whereas knowledge of tax codes may be optional. For matching work with mandatory requirements, the algorithm discards resources that do not meet hard constraints like skill, role, or location. Once the eligible set of resources is obtained, an affinity score with respect to optional factors like project, application, tools used, or account is determined from the encapsulated information in the work packet and information maintained about each worker on expertise and experience. There is an affinity score for each resource and work packet pair,  $\langle \text{res}, \text{wpk} \rangle$ .

**Time distribution of resources amongst tasks:** There may be work policies that dictate time allocation. For example, a policy may require only one task to be performed at a time whereas other policies may allow resources to perform tasks in parallel.

**Dependency among tasks within a work unit:** Typically, projects have dependencies expressed as partial ordering constraints like *start-to-finish*, *finish-to-start*, *start-to-start*, and *finish-to-finish*.

**Resource availability:** Constraints are needed to account for available time when planning for new work and may include the list of holidays for a resource.

The work to be assigned can be in-progress, starting-in-near-future, or far-in-future. The algorithm is aware of these time attributes and accordingly modifies plans. Work that is starting in the near future should undergo minimum adjustments in plan since it is psychologically important for workers to have some idea of what work is coming next in their task list. The temporal stability of the algorithm should, however, be parametrized to support cases where great dynamism is appropriate.

Robustness is also important for global service delivery so service requesters do not feel the impact of the perturbations happening within the delivery system; the system should absorb internal perturbations without hampering customer commitments and service level agreements. Indeed, if the algorithm frequently suggests many changes in the work assignment, then it may be difficult for delivery managers to make a commitment to customers and put forth a plan for each deliverable.

An interesting aspect of the constraint-based formulation is in the objectives, which have an inherent tension among them. The objectives for the real-time, large-scale work assignment are:

- 1) Work should be completed by the deadline. This should be based on the priority of the work.
- 2) Resources with the best possible skill match should be chosen for each task.
- 3) Adjust the plan to accommodate dynamic changes to work or resources such that the properties of stability and robustness are maintained.

(Note the inherent tension with the first objective, since the best resources may be engaged in other work.)

#### 3.3.1 Constraint-based program

Now let us mathematize the optimization problem. The inputs are as follows:

1. Heuristic stopping rules to limit computational complexity are included in software packages such as ILOG (<http://www-01.ibm.com/software/info/ilog/>).

- $N$ , the number of work packets,  $i \in \{1, \dots, N\}$
- $M$ , the number of resources,  $j \in \{1, \dots, M\}$
- $R$ , the number of deliverables,  $k \in \{1, \dots, R\}$
- $\sigma_{ij}$ , an input matrix where an entry is a score if location and role of  $i$  and  $j$  match and zero otherwise
- $f_i$ , the effort for packet  $i$
- $C_k$ , the committed end date for deliverable  $k$ : this may be empty for fresh ones
- $St_k$ , the input start date for deliverable  $k$
- $p \in \{1, \dots, P\}$ , an element in the set of priorities
- $\pi_p$ , the penalty of scheduling later than start date: the penalty with high priority can be set very large, e.g. it can be exponential in priority to model the objective function
- $P_{kp}$ , a matrix such that an entry is 1 if deliverable  $k$  has priority  $p$  and 0 otherwise
- $\tau \in \{1, \dots, T\}$ , the time for assignment
- $A_{j\tau}$ , a matrix derived from resource calendar
- $z_{ik}$ , is 1 if deliverable  $k$  contains the packet  $i$
- $d_{ii'}$ , the dependency type between packets  $i$  and  $i'$

The decision variables are as follows:

- $x_{ij}$ , which is true if packet  $i$  is assigned to resource  $j$ , and false otherwise
- $e_i$ , the end date of packet  $i$
- $s_i$ , the start date of packet  $i$
- $E_k$ , the end date of deliverable  $k$
- $S_k$ , the start date of packet  $k$
- $y_{j\tau}$ , which is true if resource  $j$  works at time  $\tau$

This leads to the following optimization objectives.

$$\min \sum_{k,p} \pi_p P_{kp} (E_k - St_k) \quad (\text{OBJECTIVE 1})$$

$$\max \sum_{i,j} \sigma_{ij} x_{ij} \quad (\text{OBJECTIVE 2})$$

where the two objectives can be combined into a single objective Lagrangian by subtraction. The following

constraints are also imposed.

s.t.

$$\sum_{i,j} x_{ij} \sigma_{ij} > 0 \quad (\text{MATCHING LOCATION AND ROLE})$$

$$\sum_j x_{ij} = 1 \text{ for all } i \quad (\text{ONE PACKET TO ONE RESOURCE})$$

$$\sum_j x_{ij} (e_i - s_i) \geq f_i \text{ for all } i \quad (\text{PLANNED END DATE ACCOUNTS FOR EFFORT})$$

$$\sum_j (x_{ij} x_{i'j} = 1) \Rightarrow (e_i < s'_i \vee s_i > e'_i) \text{ for all } i \neq i' \quad (\text{ONE PACKET AT A TIME})$$

$$\sum_{ii'jj'} d_{ii'} (x_{ij} x_{i'j'} = 1) \Rightarrow e_i < s'_i \quad (\text{SEQUENTIAL DEPENDENCY})$$

$$S_k \geq St_k \quad (\text{PLANNED START DATE AFTER INPUT START DATE})$$

$$C_k \neq 0 \Rightarrow E_k \leq C_k \quad (\text{END DATE OBEYS COMMITTED DATE})$$

$$E_k - S_k = \sum_{i,k} z_{ik} (e_i - s_i) \quad (\text{DELIVERABLE CONSISTS OF WORK PACKETS})$$

$$\sum_{j,\tau} (y_{j\tau} x_{ij} A_{j\tau}) = f_i \text{ for all } i$$

$$x_{ij} \in \{0, 1\}$$

This is a combinatorial optimization problem, so due to computational complexity constraints, one must use heuristic stopping criteria.

### 3.4 Dynamic Programming

With scheduling checkpoints established, a myopic form of dynamic programming that only looks a few time steps ahead is used in between [43]. Since the problem is one of assigning workers to tasks, the affinity scores defined above (Figure 3) are used as inputs to an implementation of the Hungarian method for bipartite matching [42] for optimization within stages of the dynamic programming. The Bellman principle is used for stage-by-stage optimization. Note that there are strong similarities to queuing network control [41]. The formalism is as follows.

The *decision epoch* occurs at each work packet arrival, at each work packet completion indication, and at each information update on work completion estimate, with time instances indexed as  $t = 1, 2, \dots$ . So time is event-driven. The *state space* has variables that specify how far along each worker is towards completing his/her current work packet. The *action space* is the assignment matrix  $A_t$  of work packets to workers. For each pairing between work packets and workers, the *reward* is determined by a cost  $c_{ij}(t)$  that is computed from expertise match and the various other input data. So the reward is  $\sum_{i,j:i \rightarrow j} c_{ij}(t)$ . Since the goal is throughput/utilization, this cost will typically be the time required to complete work. Due to lack of ability to preempt,  $c_{ij}(t)$  is time-dependent and has action-dependence such that when things are assigned into the future, it introduces an infinite-valued entry for  $c_{ij}(t)$ .

The *objective* is the long-term average of the cost functional:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{t=1}^T \sum_{i,j:i \rightarrow j} c_{ij}(t) \right]$$

and the *policy* is to perform optimal bipartite matching at every decision epoch (using the Hungarian method which runs in polynomial time):

$$A_t = \min_{i \rightarrow j} \sum_{i,j:i \rightarrow j} c_{ij}(t) \text{ for all } t = t_0, t_0 + 1, \dots, T$$

starting at the end horizon  $T$  and working backwards using stochastic dynamic programming, while updating the action-dependent rewards. Results are placed into each workers task list.

## 4 SIMULATION RESULTS

We designed an event-driven simulator to mimic global delivery work planning by having the work and resources instantiated in accordance with properties of real systems. The simulation engine generates deliverables with the desired details – skills required, preferred date of completion, work packets and their dependences. The system is bootstrapped with a fixed number of resources whose skill profiles match with the incoming work. Upon arrival of new deliverable requests, assignment of work packets to resources is done based on the algorithms described in Section 3. After each assignment run of the optimization engine, resources have updated work lists and calendars.

Simulation is carried out for two scenarios: SINGLE and RHC. In the SINGLE scenario, the plan once generated for deliverables does not undergo any changes. This is similar to manual project planning and scheduling processes, where planning occurs during the initial stages and does not change unless absolutely necessary. In the RHC scenario, there is re-planning at periodic intervals to ensure optimality by considering the latest set of deliverables. We define and measure two performance metrics to compare the two scenarios: percentage of deliverables meeting the deadline and percentage of resources utilized.

Figure 6 shows performance for both SINGLE and RHC. We create 25 resources with varying skills and roles. We conduct eight simulation runs with varying number of deliverables. As shown, when the number of deliverables is low (10–30) or when there are abundant resources, the percentage of tardy deliverables is the same. The percentage utilization of resources is also similar. However, as the number of deliverables increases, the percentage of tardy deliverables reduces and the resource utilization increases for the RHC scenario. Improved metrics indicate the improved efficiency of RHC as compared to SINGLE scenario. We observe up to 10% improvement in utilization when RHC is adopted.

An interesting threshold we observed was that when the number of deliverables is very large (90 or 1080 work

packets), some of the resources reached utilization of  $> 95\%$ . This acted as a limit on further improvement due to RHC. However, with a different set of work definitions, it is possible to achieve up to 15% improvement.

Since the work definition used for simulation was taken from real projects, the results provide significant insights into the real system deployment, discussed next.

## 5 SYSTEM DEPLOYMENTS

In this section, we provide two brief case studies of some deployments of the system described in this work.

### 5.1 Tire Manufacturer

*Client 1:* World leader in manufacturing tires and related products. Project involved integration of disparate systems into cohesive Order-To-Cash (OTC) functionality across the client's enterprise.

*Challenge 1:* Client needed to address specific business challenges to remain competitive. In particular, data across several legacy systems could not be shared efficiently, the technology environment could not support real-time or predictive information analysis, the supply chain could not be viewed and managed holistically, and there were high costs related to maintaining multiple legacy systems.

*Solution 1:* The service engagement therefore was to develop a service-oriented architecture based on an enterprise service bus, so as to integrate various disparate systems and technologies (mainframe, DRP, SAP, Highjump, etc.) as well as legacy systems. There was need to support all transformation logic at a centralized place in middleware and provide a standard method for extracting enterprise resource planning data into other systems for analysis.

The solution was implemented using multiple IBM Global Delivery locations across the U.S., India, and China. The multifarious, interdependent, and high-volume work was done by following the delivery methods and coordination mechanisms described herein. Indeed, as part of delivery over 210 technology interfaces were identified and developed.

*Value 1:* In leveraging global assets to improve productivity, the cognitive approach reduced coordination costs and allowed 10% reduction in effort hours. For the client, this increased business efficiency through integration of business processes, provided legacy synchronization of master and transactional data across divisions, and improved the client's responsiveness to changing business needs.

### 5.2 Telecommunications Company

*Client 2:* Major telecommunications company which offers the local exchange carrier for telephone and DSL Internet services in most of Canada. Project involved enabling a business model that improved performance

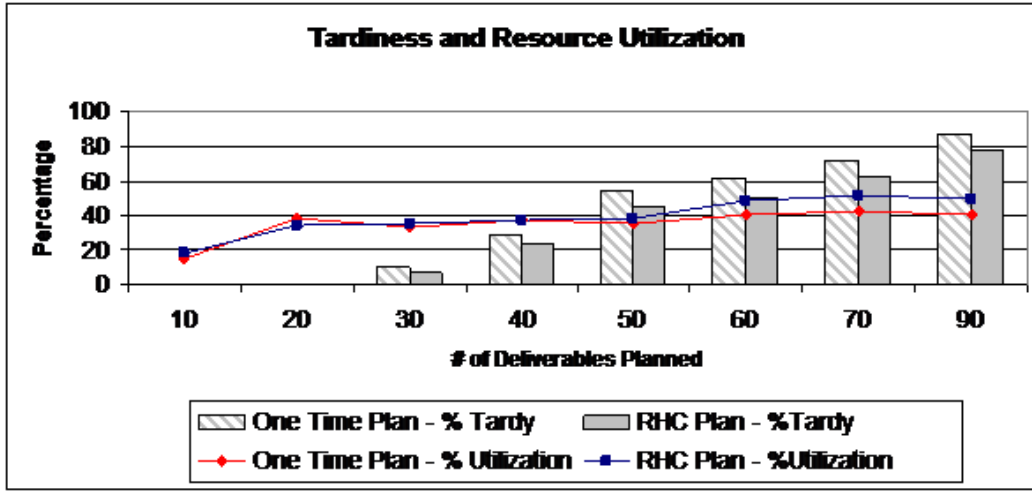


Fig. 6. Comparing the performance of RHC coordination with typical human-oriented coordination.

and reduced cost in performing data extraction, transformation and loading (ETL) via architecture standards and best practices for sustainable productivity improvements.

*Challenge 2:* Client architecture met basic requirement of data movement but was unstructured and inefficient. The gaps in the architecture are also pervasive at the implementation level such as no standardized components, redundancy, and inflexible solutions. Additionally, the client was unable to provide detailed implementation specifications due to their contractual agreements with their customers and vendors.

*Solution 2:* Based on initial analysis, IBM team developed client/environment specific recommendations and generic best practices based on the cognitive coordination approach for Data Stage and Tera Data. The engagement team created and presented communication plans catering to different groups both within the client organization and with vendors to enable ETL 2.0 recommendations. The business analytics and optimization team performed analysis of various applications and downstream data integration methodologies to provide recommendations and solutions for efficiency. The team also recommended multiple approaches for scalability and maintenance by leveraging data virtualization in combination with data integration.

*Value 2:* The cognitive coordination model helped in reducing the overall cost for the project through a shared delivery model, usage of assets and accelerators and collaboration with technical subject matter experts. The estimation model provided a 10% reduction in the overall effort level thereby helping the client meet tight timelines and budget. The reduction was achieved through leveraging reusable components, and parallel processing. Also, using the encapsulated templates, checklists and best practices, the project was able to achieve a significant reduction (more than 50%) in the number of defects. It successfully overcame the challenges of work-

ing across different time zones and multiple languages and overlapping waves to ensure smooth delivery. Finally, the concept of *golden data client* was used to build production quality data ahead of time and get business user commitment and ownership for data.

## 6 CONCLUSION

Sociotechnical systems for delivering services are subject to various forms of uncertainty. Indeed, “uncertainty is what typifies projects. It’s the nature of the beast” [44]. Though always present, this inherent uncertainty is becoming more noticeable as inefficiencies are being squeezed out of service organizations. As has been noted, “after years of optimizing supply chains, outsourcing, automation, and stripping costs and inefficiencies out of the back office, most employees spend very little of their day working on regularized activities. What they do is they manage exceptions to processes” [45].

These issues are magnified in the *global* service delivery context, where manual coordination procedures have become inefficient in dealing with uncertainties in a scalable manner. Formal coordination mechanisms that measure system state and take actions to respond are becoming crucial.

In this work, we have reported on our experience in coordinating a large-scale sociotechnical system for global service delivery. Using a cognitive coordination framework, a Markov decision process formulation, and computationally-implementable receding horizon control algorithms, we have developed a middleware deployment that achieves 10–15% improvement over existing coordination approaches. These improvements are measured not only in realistic simulation studies, but also in client project deployments. The basic frameworks, formulations, algorithms, and technologies can serve as the basis for other similar problems of coordination.



## ACKNOWLEDGMENT

The authors thank Rong Liu and Bikram Sengupta for discussions, and colleagues in IBM Global Business Services for deploying the system described herein.

## REFERENCES

- [1] L. R. Varshney and D. V. Oppenheim, "Coordinating global service delivery in the presence of uncertainty," in *Proc. 12th Int. Research Symp. Service Excellence Manage. (QUIS12)*, Jun. 2011, pp. 1004–1014.
- [2] D. Bollier, *The Future of Work: What It Means for Individuals, Businesses, Markets and Governments*. Washington, DC: The Aspen Institute, 2011.
- [3] T. W. Malone, R. J. Laubacher, and T. Johns, "The age of hyper-specialization," *Harvard Bus. Rev.*, vol. 89, no. 7/8, pp. 56–65, July-Aug. 2011.
- [4] S. J. Palmisano, "The globally integrated enterprise," *Foreign Aff.*, vol. 85, no. 3, pp. 127–136, May-June 2006.
- [5] G. S. Becker and K. M. Murphy, "The division of labor, coordination costs, and knowledge," *Quart. J. Econ.*, vol. 107, no. 4, pp. 1137–1160, Nov. 1992.
- [6] M. Ehret and J. Wirtz, "Division of labor between firms: Business services, non-ownership-value and the rise of the service economy," *Service Sci.*, vol. 2, no. 3, pp. 136–145, Fall 2010.
- [7] G. J. Stigler, "The division of labor is limited by the extent of the market," *J. Polit. Econ.*, vol. 59, no. 3, pp. 185–193, Jun. 1951.
- [8] H. Mintzberg, *Mintzberg on Management*. New York: Free Press, 1989.
- [9] D. C. Gumm, "Distribution dimensions in software development projects: A taxonomy," *IEEE Softw.*, vol. 23, no. 5, pp. 45–51, Sept-Oct. 2006.
- [10] G. O. Wiredu, "A framework for the analysis of coordination in global software development," in *Proc. 2006 Int. Workshop Global Softw. Dev. Practitioner*, May 2006, pp. 38–44.
- [11] R. Gulati, F. Wohlgezogen, and P. Zhelyazkov, "The two facets of collaboration: Cooperation and coordination in strategic alliances," *Acad. Manage. Ann.*, vol. 6, no. 1, pp. 531–583, Jun. 2012.
- [12] IBM Global Business Services, "Application assembly optimization: A new approach to global delivery," Aug. 2009.
- [13] D. M. Upton and V. A. Fuller, "Wipro technologies: The factory model," Harvard Business School: 9-606-021, Oct. 2005.
- [14] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "Representations of space and time in the maximization of information flow in the perception-action loop," *Neural Comput.*, vol. 19, no. 9, pp. 2387–2432, Sep. 2007.
- [15] S. Haykin, "Cognitive dynamic systems: Radar, control, and radio," *Proc. IEEE*, vol. 100, no. 7, pp. 2095–2103, Jul. 2012.
- [16] —, "Cognitive dynamic systems," *Proc. IEEE*, vol. 102, no. 4, pp. 414–416, Apr. 2014.
- [17] S. Haykin and J. M. Fuster, "On cognitive dynamic systems: Cognitive neuroscience and engineering learning from each other," *Proc. IEEE*, vol. 102, no. 4, pp. 608–628, Apr. 2014.
- [18] D. Oppenheim, S. Bagheri, K. Ratakonda, and Y.-M. Chee, "Coordinating distributed operations," in *Service-Oriented Computing*, ser. Lecture Notes in Computer Science, E. M. Maximilien, G. Rossi, S.-T. Yuan, H. Ludwig, and M. Fantinato, Eds. Berlin: Springer, 2011, vol. 6568, pp. 213–224.
- [19] J. Mattingley, Y. Wang, and S. Boyd, "Receding horizon control," *IEEE Control Syst. Mag.*, vol. 31, no. 3, pp. 52–65, Jun. 2011.
- [20] N. Desai, A. K. Chopra, and M. P. Singh, "Amoeba: A methodology for modeling and evolving cross-organizational business processes," *ACM Trans. Softw. Eng. Methodol.*, vol. 19, no. 2, p. 6, Oct. 2009.
- [21] D. V. Oppenheim, S. Bagheri, K. Ratakonda, and Y.-M. Chee, "Agility of enterprise operations across distributed organizations: A model of cross enterprise collaboration," in *Proc. SRII Global Conf. 2011*, Mar. 2011.
- [22] F. Leymann and D. Roller, *Production Workflow: Concepts and Techniques*. Upper Saddle River, NJ: Prentice Hall, 2000.
- [23] A. Nigam and N. S. Caswell, "Business artifacts: An approach to operational specification," *IBM Syst. J.*, vol. 42, no. 3, pp. 428–445, 2003.
- [24] D. V. Oppenheim, L. R. Varshney, and Y.-M. Chee, "Work as a service," in *Service-Oriented Computing*, ser. Lecture Notes in Computer Science, G. Kappel, Z. Maamar, and H. R. Motahari-Nezhad, Eds. Berlin: Springer, 2011, vol. 7084, pp. 669–678.
- [25] —, "Work as a service," in *Advanced Web Services*, A. Bouguet-taya, Q. Z. Sheng, and F. Daniel, Eds. Springer, 2014, pp. 409–430.
- [26] R. Vaculin, Y.-M. Chee, D. V. Oppenheim, and L. R. Varshney, "Work as a service meta-model and protocol for adjustable visibility, coordination, and control," in *Proc. SRII Global Conf. 2012*, Jul. 2012, pp. 90–99.
- [27] T. S. Kuhn, "The function of measurement in modern physical science," *Isis*, vol. 52, no. 2, pp. 161–193, Jun. 1961.
- [28] D. V. Oppenheim, Y.-M. Chee, and L. R. Varshney, "Allegro: A metrics framework for globally distributed service delivery," in *Proc. SRII Global Conf. 2012*, Jul. 2012, pp. 461–469.
- [29] J. Wang, K. R. Varshney, A. Mojsilović, D. Fang, and J. H. Bauer, "Expertise assessment with multi-cue semantic information," in *Proc. 2013 IEEE Int. Conf. Serv. Oper. Logist. Inform. (SOLI)*, Jul. 2013, pp. 534–539.
- [30] Y. Li and K. Katircioglu, "Measuring and applying service request effort data in application management services," in *Proc. 2013 IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2013, pp. 352–359.
- [31] R. Liu, S. Agarwal, R. R. Sindhgatta, and J. Lee, "Accelerating collaboration in task assignment using a socially enhanced resource model," in *Business Process Management*, ser. Lecture Notes in Computer Science, F. Daniel, J. Wang, and B. Weber, Eds. Berlin: Springer, 2013, vol. 8094, pp. 251–258.
- [32] S. Graham, G. Baliga, and P. R. Kumar, "Abstractions, architecture, mechanisms, and a middleware for networked control," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1490–1503, Jul. 2009.
- [33] S. Brusoni and A. Prencipe, "Unpacking the black box of modularity: Technologies, products and organizations," *Ind. Corporate Change*, vol. 10, no. 1, pp. 179–205, Mar. 2001.
- [34] K. Foss, "Organizing technological interdependencies: a coordination perspective on the firm," *Ind. Corporate Change*, vol. 10, no. 1, pp. 151–178, Mar. 2001.
- [35] C. Y. Baldwin, "Where do transactions come from? modularity, transactions, and the boundaries of firms," *Ind. Corporate Change*, vol. 17, no. 1, pp. 155–195, Feb. 2008.
- [36] T. W. Malone and K. Crowston, "The interdisciplinary study of coordination," *ACM Comput. Surv.*, vol. 26, no. 1, pp. 87–119, Mar. 1994.
- [37] V. Bellotti, B. Dalal, N. Good, P. Flynn, D. G. Bobrow, and N. Ducheneaut, "What a to-do: studies of task management towards the design of a personal task list manager," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. (CHI 2004)*, Apr. 2004, pp. 735–742.
- [38] R. A. Howard, *Dynamic Programming and Markov Processes*. Cambridge, MA: MIT Press, 1960.
- [39] Y. Naveh, Y. Richter, Y. Altshuler, D. L. Gresh, and D. P. Connors, "Workforce optimization: Identification and assignment of professional workers using constraint programming," *IBM J. Res. Develop.*, vol. 51, no. 3/4, pp. 263–279, May 2007.
- [40] S. Asaf, H. Eran, Y. Richter, D. P. Connors, D. L. Gresh, J. Ortega, and M. J. Mcinnis, "Applying constraint programming to identification and assignment of service professionals," in *Principles and Practice of Constraint Programming*, ser. Lecture Notes in Computer Science, D. Cohen, Ed. Berlin: Springer, 2010, vol. 6308, pp. 24–37.
- [41] P. Tehrani and Q. Zhao, "Multichannel scheduling and its connection to queueing network control problem," in *Proc. Mil. Commun. Conf. (MILCOM 2010)*, Nov. 2010, pp. 482–486.
- [42] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist. Q.*, vol. 2, no. 1-2, pp. 83–97, Mar. 1955.
- [43] R. Bellman, *Dynamic Programming*. Princeton: Princeton University Press, 1957.
- [44] E. M. Goldratt, *Critical Chain*. North River Press, 1997.
- [45] D. Tapscott and A. D. Williams, *Wikinomics: How Mass Collaboration Changes Everything*, expanded ed. New York: Portfolio Penguin, 2006.